This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2558446, IEEE Access

# Protection of Big Data Privacy

Abid Mehmood, Iynkaran Natgunanathan, Yong Xiang, *Senior Member, IEEE*,
Guang Hua, *Member, IEEE*, and Song Guo, *Senior Member, IEEE*

***Abstract*—In recent years big data has become a hot research topic. The increasing amount of big data also increases the chance of breaching the privacy of individuals. Since big data requires high computational power and large storage, distributed systems are used. As multiple parties are involved in these systems, the risk of privacy violation is increased. There has been a number of privacy-preserving mechanisms developed for privacy protection at different stages (e.g., data generation, data storage, and data processing) of a big data life cycle. The goal of this paper is to provide a comprehensive overview of the privacy preservation mechanisms in big data and present the challenges for existing mechanisms. Specifically, in this paper we illustrate the infrastructure of big data and the state-of-the-art privacy-preserving mechanisms in each stage of the big data life cycle. Furthermore, we discuss the challenges and future research directions related to privacy preservation in big data.**

***Index Terms*—Big data, privacy, data auditing, big data storage, big data processing.**

## I. Introduction

Due to recent technological development, the amount of data generated by social networking sites, sensor networks, Internet, healthcare applications, and many other companies, is drastically increasing day by day. All the huge amount of data generated from different sources in multiple formats with very high speed is referred as big data. Big data has become a very active research area for last couple of years. The data generation rate is growing so rapidly that it is becoming extremely difficult to handle it using traditional methods or systems [1]. Meanwhile, big data could be structured, semi-structured, or unstructured, which adds more challenges when performing data storage and processing tasks. Therefore, to this end, we need new ways to store and analyse data in real time. Big data, if captured and analyzed in a timely manner, can be converted into actionable insights which can be of significant value. It can help businesses and organizations to improve the internal decision making power and can create new opportunities through data analysis. It can also help to promote the scientific research and economy by transforming traditional business models and scientific values [2].

Big data can be defined in various ways. For the scope of this paper we use the definition given by International Data Corporation (IDC) in [3]. In [3], the term big data is defined as "a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis". Based on this definition, the properties of big data are reflected by 3 V's, which are, volume, velocity and variety, as shown in Fig. 1. Volume refers to the amount of data generated. With the emergence of social networking sites, we have seen a dramatic increase in the size of the data. The rate at which new data are generated is often characterized as velocity. A common theme of big data is that the data are diverse, i.e., they may contain text, audio, image, or video etc. This diversity of data is denoted by variety.

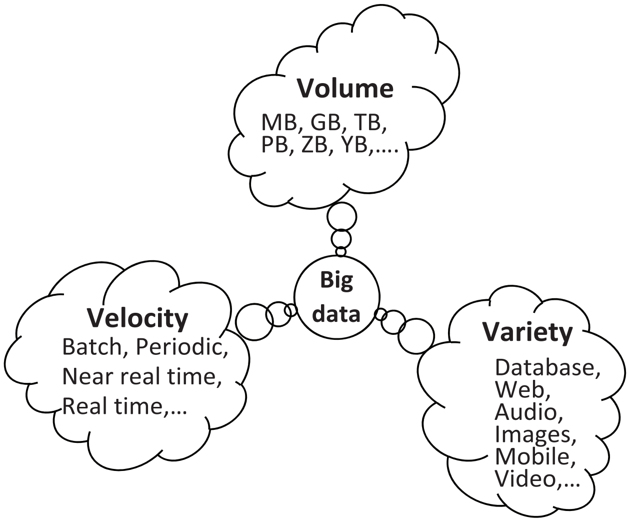

Fig. 1. Illustration of the 3 V's of big data.

Despite big data could be effectively utilized for us to better understand the world and innovate in various aspects of human endeavors, the exploding amount of data has increased potential privacy breach. For example, Amazon and Google can learn our shopping preferences and browsing habits. Social networking sites such as Facebook store all the information about our personal life and social relationships. Popular video sharing websites such as YouTube recommends us videos based on our search history. With all the power driven by big data, gathering, storing and reusing our personal information for the purpose of gaining commercial profits, have put a threat to our privacy and security. In 2006, AOL released 20 million search queries for 650 users by removing the AOL id and IP address for research purposes. However, it took researchers only couple of days to re-identify the users. Users' privacy may be breached under the following circumstances [4]:

- Personal information when combined with external datasets may lead to the inference of new facts about the users. Those facts may be secretive and not supposed

A. Mehmood, I. Natgunanathan, and Y. Xiang are with the School of Information Technology, Deakin University, Victoria, Australia (tel: +61 3 92517740, fax: +61 3 92446831, e-mail: {amehm, iynkaran, yxiang}@deakin.edu.au).

G. Hua is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: ghua@ntu.edu.sg).

S. Guo is with the Department of Computer Science and Engineering, The University of Aizu, Japan (e-mail: sguo@u-aizu.ac.jp).

to be revealed to others.

- Personal information is sometimes collected and used to add value to business. For example, individual's shopping habits may reveal a lot of personal information.
- The sensitive data are stored and processed in a location not secured properly and data leakage may occur during storage and processing phases.

In order to ensure big data privacy, several mechanisms have been developed in recent years. These mechanisms can be grouped based on the stages of big data life cycle, i.e., data generation, storage, and processing. In data generation phase, for the protection of privacy, access restriction and falsifying data techniques are used. While access restriction techniques try to limit the access to individuals' private data, falsifying data techniques alter the original data before they are released to a non-trusted party. The approaches to privacy protection in data storage phase are mainly based on encryption techniques. Encryption based techniques can be further divided into attribute based encryption (ABE), Identity based encryption (IBE), and storage path encryption. In addition, to protect the sensitive information, hybrid clouds are used where sensitive data are stored in private cloud. The data processing phase includes privacy preserving data publishing (PPDP) and knowledge extraction from the data. In PPDP, anonymization techniques such as generalization and suppression are used to protect the privacy of data. Ensuring the utility of the data while preserving the privacy is a great challenge in PPDP. In the knowledge extracting process, there exist several mechanisms to extract useful information from large-scale and complex data. These mechanisms can be further divided into clustering, classification and association rule mining based techniques. While clustering and classification split the input data into different groups, association rule mining based techniques find the useful relationships and trends in the input data.

Protecting privacy in big data is a fast growing research area. Although some related papers have been published but only few of them are survey/review type of papers [2], [5]. Moreover, while these papers introduced the basic concept of privacy protection in big data, they failed to cover several important aspects of this area. For example, neither [2] nor [5] provides detailed discussions regarding big data privacy with respect to cloud computing. Besides, none of the papers discussed future challenges in detail.

In this paper, we will give a comprehensive overview of the state-of-the-art technologies to preserve privacy of big data at each stage of big data life cycle. Moreover, we will discuss privacy issues related to big data when they are stored and processed on cloud, as cloud computing plays very important role in the application of big data. Furthermore, we will discuss about potential research directions. The remainder of this paper is organized as follows. The infrastructure of big data and issues related to privacy of big data because of the underlying structure of cloud computing will be discussed in section II. Privacy issues related to data generation phase will be discussed in section III. Issues related to privacy during data storage and data processing phase will be discussed in sections IV and V, respectively. Finally future research directions are identified and discussed in section VI.

## II. Infrastructure of big data

To handle different dimensions of big data in terms of volume, velocity, and variety, we need to design efficient and effective systems to process large amount of data arriving at very high speed from different sources. Big data has to go through multiple phases during its life cycle, as shown in Fig. 2. Data are distributed nowadays and new technologies are being developed to store and process large repositories of data. For example, cloud computing technologies, such as Hadoop MapReduce, are explored for big data storage and processing.

In this section we will explain the life cycle of big data. In addition, we will also discuss how big data are leveraging from cloud computing technologies and drawbacks associated with cloud computing when used for storage and processing of big data.

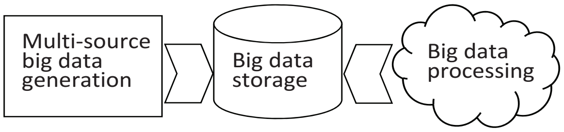

Fig. 2. Illustration of big data life cycle.

### A. *Life cycle of big data*

- Data generation: Data can be generated from various distributed sources. The amount of data generated by humans and machines has exploded in the past few years. For example, everyday 2.5 quintillion bytes of data are generated on the web and 90 percent of the data in the world is generated in the past few years. Facebook, a social networking site alone is generating 25TB of new data everyday. Usually, the data generated is large, diverse and complex. Therefore, it is hard for traditional systems to handle them. The data generated are normally associated with a specific domain such as business, Internet, research, etc.
- Data storage: This phase refers to storing and managing large-scale data sets. A data storage system consists of two parts i.e., hardware infrastructure and data management [6]. Hardware infrastructure refers to utilizing information and communications technology (ICT) resources for various tasks (such as distributed storage). Data management refers to the set of software deployed on top of hardware infrastructure to manage and query large scale data sets. It should also provide several interfaces to interact with and analyze stored data.
- Data processing: Data processing phase refers basically to the process of data collection, data transmission, preprocessing and extracting useful information. Data collection is needed because data may be coming from different diverse sources i.e., sites that contains text, images and videos. In data collection phase, data are acquired from specific data production environment using dedicated data collection technology. In data transmission phase, after

collecting raw data from a specific data production environment we need a high speed transmission mechanism to transmit data into a proper storage for various type of analytic applications. Finally, the pre-processing phase aims at removing meaningless and redundant parts of the data so that more storage space could be saved.
The excessive data and domain specific analytical methods are used by many application to derive meaningful information. Although different fields in data analytics require different data characteristics, few of these fields may leverage similar underlying technology to inspect, transform and model data to extract value from it. Emerging data analytics research can be classified into the following six technical areas: structured data analytics, text analytics, multimedia analytics, web analytics, network analytics, and mobile analytics [6].

### B. Cloud computing and big data

Big data need massive computation and storage, which brings in the need for cloud computing. Cloud computing is driving enterprises and businesses to adopt cloud, because of many advantages it is offering, such as cost saving and scalability. It also offers huge processing power and storage capability. Technologies used in cloud computing like virtualization, distributed storage and processing have made it possible to perform tasks that had been considered difficult in conventional system. However, on the other hand, could computing also results in serious cloud specific privacy issues. People hesitate to transfer their private or sensitive data to the cloud unless they are sure that their data will be secure on the cloud. There are some challenges for building a trustworthy and secure big data storage and processing system on cloud which are as follows [7].

- Outsourcing: To reduce the capital and operational expenditure, organizations nowadays prefer to outsource their data to the cloud. However, outsourcing data to cloud also means that the customers will lose physical control on their data. The loss of control over the data has become one of the main cause of cloud insecurity. The insecurity can cause serious damage to the privacy of cloud computing customer. These issues can be addressed by providing secure computing environment and data storage. In addittion to that, outsourced data should also be verifiable to customers in terms of confidentiality and integrity.
- Multi-tenancy: Virtualization has made it possible to share the same cloud platform by multiple customers. The data that belong to different cloud users may be placed on the same physical storage by some resource allocation policy. In such an environment, it is relatively easy for a malicious user to illegally access data which do not belong to him. A series of issues may occur in such an environment, such as data breach and computation breach. Due to that, it is very important to design mechanisms to deal with potential privacy and security risks.
- Massive computation: Due to the capability of cloud computing for handling massive data storage and intense computations, traditional mechanisms to protect individual's privacy are not sufficient.

## III. Privacy in data generation phase

Data generation can be classified into active data generation and passive data generation. Active data generation means that the data owner is willing to provide the data to a third party, while passive data generation refers to the situations that the data are generated by data owner's online activity (e.g., browsing) and the data owner may not even be aware of that the data are being collected by a third party. The major challenge for data owner is that how can he protect his data from any third party who may be willing to collect them. The data owner wants to hide his personal and sensitive information as much as possible and is concerned about how much control he could have over the information. We can minimize the risk of privacy violation during data generation by either restricting the access or by falsifying data [5].

### A. Access restriction

If the data owner thinks that the data may reveal sensitive information which is not supposed to be shared, he can simply refuse to provide such data. For that, the data owner has to adopt effective access control methods so that the data can be prevented from being stolen by some third party. If the data owner is providing the data passively, some measures could be taken to ensure privacy, such as anti-tracking extensions, advertisement/script blockers and encryption tools [5]. By using these tools, one can effectively limit the access to sensitive data. For the ease of use, most of these tools are designed as browser extensions.

In addition to these tools, there are some alternative means, such as to use anti-malware and anti-virus software to protect the data stored digitally on their computer or laptop. These tools can help to protect user's personal data by limiting the access. Though there is no guarantee that one's sensitive data are completely protected from untrustworthy sources, making it a habit of clearing online traces of one's activity by using security tools can significantly reduce the risk.

### B. Falsifying data

In some circumstances, it is not possible to prevent access of sensitive data. In that case, data can be distorted using certain tools before the data are fetched by some third party. If the data are distorted, the true information cannot be easily revealed. The following techniques are used by the data owner to falsify the data [5].

- A tool *Socketpuppet* is used to hide online identity of individual by deception. Individual's true activities online are concealed by creating a false identity and pretending to be someone else. By using multiple *Socketpuppets*, the data belonging to one specific individual will be deemed as belonging to different individuals. In that way the data collector will not have enough knowledge to relate different *socketpuppets* to one individual. Hence, the users true activities are unknown to others and the private information cannot be discovered easily.

- Certain security tools can be used to mask individuals identity, such as *MaskMe*. It allows users to create aliases of their personal information such as email address or credit card number. The data owner can use these masks whenever information is needed. This is especially useful when the data owner needs to provide the credit card details during online shopping.

## IV. Privacy in data storage phase

Storing high volume data is not a big challenge due to the advancement in data storage technologies such as the boom in cloud computing. However, securing the data is very challenging. If the big data storage system is compromised, it can be very harmful as individuals' personal information can be disclosed. Therefore, we need to ensure that the stored data are protected against such threats. In modern information systems, data centres play an important role of performing complex commutations and retrieving large amount of data. In distributed environment, an application may need several datasets from different data centres and therefore face the challenge of privacy protection.

The conventional security mechanisms to protect data can be divided into four categories. They are file level data security schemes, database level data security schemes, media level security schemes and application level encryption schemes [8]. The conventional mechanism to protect data security [9] and privacy [10]-[11] for existing storage storage architectures (i.e., direct attached storage, network attached storage and storage area network) [12] have been a very hot research area but may not be directly applicable to big data analytics platform. In response to the 3V's nature of the big data analytics, the storage infrastructure should be scalable. It should have the ability to be configured dynamically to accommodate diverse applications. One promising technology to address these requirements is storage virtualization, enabled by the emerging cloud computing paradigm [13]. Storage virtualization is process in which multiple network storage devices are combined into what appears to be a single storage device. However, using a cloud service offered by cloud provider means that the organization's data will be outsourced to a third party such as cloud provider. This could affect the privacy of the data. Therefore, in this paper we will limit our discussions to privacy of data when stored on cloud.

### A. Approaches to privacy preservation storage on cloud

When data are stored on cloud, data security mainly has three dimensions, confidentiality, integrity and availability [7]. The first two are directly related to privacy of the data i.e., if data confidentiality or integrity is breached it will have a direct effect on users privacy. Therefore we will also discuss privacy issues related to confidentiality and integrity of data in this section.

A basic requirement for big data storage system is to protect the privacy of an individual. There are some existing mechanisms to fulfil that requirement. For example, a sender can encrypt his data using pubic key encryption (PKE) in such a way that only the valid recipient can decrypt the data. The approaches to preserve the privacy of the user when data are stored on the cloud are as follows.

*1) Attribute based encryption:* ABE [14]-[15] is an encryption technique which ensures end to end big data privacy in cloud storage system. In ABE access polices are defined by data owner and data are encrypted under those policies. The data can only be decrypted by the users whose attributes satisfy the access policies defined by the data owner. When dealing with big data one may often need to change data access policies as the data owner may have to share it with different organizations. The current attribute based access control schemes [16]-[17] do not consider policy updating. The policy updating is a very challenging task in attribute based access control systems. The reason for that is once the data are outsourced to the cloud, the data owner would not keep the local copy in the system. If the data owner wants to update the policy, he has to transfer the data back to the local system, re-encrypt the data under new policy and store it back on the cloud server. This process has got very high communication overhead and high computational cost. To solve the problem of policy updating, recently Yang *et al.* [18] proposed a secure and verifiable policy updating outsourcing method. In [18], data owner does not need to retrieve all the data and re-encrypt it. Instead the data owner can send the queries to cloud to update the policy, and the cloud server can update the policy directly without decrypting the data.

*2) Identity based encryption:* IBE is an alternative to PKE which is proposed to simplify key management in a certificate-based public key infrastructure (PKI) by using human identities like email address or IP address as public keys. To preserve the anonymity of sender and receiver, the IBE [19] scheme was proposed. By employing these primitives, the source and the destination of data can be protected privately. Encryption scheme like IBE and ABE does not support the update of ciphertext receiver. There are some approaches to updating the ciphertext recipient. For instance, data owner can employ the decrypt then re-encrypt mode. However, if data are large as it is mostly the case when dealing with big data, the decryption and re-encryption can be very time consuming and costly because of computation overhead. Moreover, in this mode, data owner has to be online all the time. Another approach to updating ciphertext receiver is to delegate this task to a trusted third party with the knowledge of decryption key of the data owner. This approach has few drawbacks like the scheme relies on the fully trust of the third party and also the anonymity of the ciphertext receiver cannot be achieved as the third party needs to know the information about the receipt to proceed the re-encryption. Mambo and Okamoto [20] introduced proxy-re encryption (PRE) which was further defined in [21]. PRE is proposed to handle the problem of data sharing between different receipts. In [20], a semi trusted third party transforms a ciphertext intended for one user into a ciphertext of the same message intended for another user without leaking any knowledge about the message or the decryption keys. The workload of data owner is now transferred to the proxy and the proxy does not have to be online all the time.

In [22], proxy re-encryption is employed in the IBE setting. In [23], anonymous identity based proxy re-encryption

(IBPRE) was introduced but the work only supports one time ciphertext receiver update, while in practice multiple receivers update is desirable. On the other hand, the work provides an all or nothing share mode that limits the flexibility. Liang *et al.* [24] proposed an anonymous identity based proxy re-encryption scheme with the following properties: the identity information of sender and receiver is anonymous and the ciphertext receiver can be updated multiple times, with the possibility of conditional fine grained sharing of ciphertext.

*3) Homomorphic encryption:* Public cloud is more vulnerable to privacy breaches because of multi-tenancy and virtualization. The cloud users may share the same physical space and in such a scenario the chances of data leakage are very high. One way to protect the data on cloud is to encrypt the data and store them on cloud and allow the cloud to perform computations over encrypted data. Fully homomorphic encryption is the type of encryption which allows functions to be computed on encrypted data [25]. Given only the encryption of a message, one can obtain an encryption of a function of that message by computing directly on the encryption. Homomorphic encryption provides full privacy but it comes at the cost of computational complexity and sometimes very hard to implement with existing technologies. A comparison of different encryption schemes is shown in Table I.

*4) Storage path encryption:* Recently Cheng *et al.* [8] proposed a scheme for secure storage of big data on clouds. In the proposed scheme, the big data are first separated into many sequenced parts and then each part is stored on a different storage media owned by different cloud storage providers. To access the data, different parts are first collected together from different data centres and then restored into original form before it is presented to the data owner. In this scheme the big data stored on the cloud is classified into public data and confidential data. There are no extra security requirements for public data and each tenant can access the data freely. In contrast, confidential data are always kept secure and inaccessible to irrelevant individual and organizations. A trapdoor function has been incorporated in this scheme. It is a function which is easy to compute in one way and difficult to compute in the opposite direction without some additional information. The trapdoor functions are used widely in cryptographic applications. In the proposed scheme instead of encrypting the whole big data, only the storage path is encrypted which is called the cryptographic virtual mapping of big data. For some special applications, the proposed scheme also encrypts some part of data which are considered confidential. In order to improve the availability and robustness of the big data, the scheme will store the copies for each piece of data on cloud storage, so that when the information or data part is lost we can try to find another copy. The owner of the big data will keep the storage index information [8].

*5) Usage of Hybrid clouds:* According to the national institute of standards and technology (NIST), the cloud can be deployed by the following three models [13]: private clouds (owned and accessed only by the providing enterprise), public cloud (available and accessible by all service subscribers), and hybrid clouds (a combination of public and private cloud).

Private clouds are inherently trustworthy and secure but there are some limitations which hamper the private clouds for the processing and storage of big data [26]. The first limitation is scalability. Building a highly scalable private cloud requires a large capital investment. It becomes very difficult to accurately plan private cloud capacity when the volume, velocity, and variety of the data are constantly changing. The second limitation is unavailability of analytical models and software frameworks required to manage heterogeneous data. The third limitation is on data sharing. Sometimes, data sharing should be available among authorized collaborators who do not have access or reside outside of private cloud. However, due to security concerns, this is not always possible. On the other hand, public cloud support scalability and easy sharing of data. However public clouds are more prone to security and privacy attacks because of the multi-tenancy of virtual machines and data.

Hybrid cloud is the combination of public cloud and private cloud. It brings together the inherent features of public clouds i.e., scalability, processing power etc. and private clouds i.e., security and provides potential research opportunities for processing and storage of big data. In [26], hybrid clouds have been deployed for the privacy preserving processing and storage of big data. We can take advantage of hybrid cloud by separating sensitive data from non-sensitive data and storing them in trusted private cloud and un-trusted public cloud respectively [27]. However, this approach has a drawback because if we adopt this approach directly, all the sensitive data have to be stored in private cloud, which would require a lot of storage in private cloud. Most users want to minimize the storage and computation in private cloud, and let public cloud do most of the storage and computation. The authors in [27] have presented a scheme to reduce the communication overhead between private and public cloud besides achieving privacy protection by using hybrid cloud. Specifically, data privacy is achieved by dividing the image into pieces and then shuffling these pieces directly. Each pixel of every block is mapped into another value via random one to one mapping function. The transformed image is stored on public cloud whereas the mapping function to recover the image is stored on private cloud.

## B. Integrity verification of big data storage

When cloud computing is used for big data storage, data owner loses control over data. The outsourced data are at risk as cloud server may not be fully trusted. The data owner needs to be strongly convinced that the cloud is storing data properly according to the service level contract. One way to ensure privacy to the cloud user is to provide the system with the mechanism to let data owner verify that his data stored on the cloud is intact. Therefore data integrity verification is of critical importance. Table II compares different integrity verification schemes discussed in this paper. Numerous research problems have been studied over the past decade [28]-[37]. The integrity of data storage in traditional systems can be verified through number of ways i.e., Reed-Solomon code, checksums, trapdoor hash functions, message authentication

TABLE I
COMPARISON OF ENCRYPTION SCHEMES

| Encryption scheme | Features | Limitations |
|---|---|---|
| Identity based encryption | • Access control is based on the identity of a user<br>• Complete access over all resources | • Time consuming in large environment<br>• Granular access control is hard to implement<br>• Changing ciphertext receiver is not possible<br>• Data to be processed must be downloaded and decrypted |
| Attribute based encryption | • Access control is based on user's attribute<br>• More secure and flexible as granular access control is possible | • Computational overhead in handling different user categories<br>• Updating ciphertext receiver is not possible<br>• Data to be processed must be downloaded and decrypted |
| Proxy re-encryption | • Can be deployed in IBE or ABE scheme settings<br>• Updating Ciphertext receiver is possible | • Computational overhead<br>• Data to be processed must be downloaded and decrypted |
| Homomorphic encryption | • Computations are performed on the encrypted data<br>• Very secure | • Computational overhead is very high |

code (MAC), and digital signatures etc. To verify the integrity of the data stored on cloud, one straight forward approach is to retrieve all the data from the cloud. However, the great volume of big data makes it very inefficient to consider time consumption and communication overhead. To address this problem, researchers have developed schemes to verify the integrity of data without having to retrieve the data from cloud [28]-[29]. In integrity verification scheme, the cloud server can only provide the valid proof of integrity of data when all the data are intact. It is highly recommended that the integrity verification should be conducted regularly to provide highest level of data protection [28]. In the following, we will discuss the framework of integrity verification, followed by popular integrity verification schemes for dynamic data. Note that the data in most big data applications are dynamic in nature. Fig. 3 explains the basic framework of integrity verification schemes.

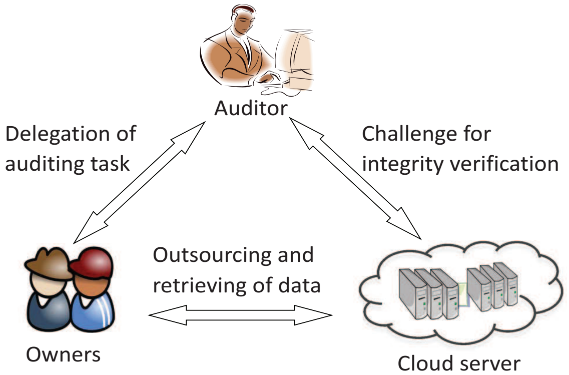
Auditor
Delegation of auditing task
Challenge for integrity verification
Outsourcing and retrieving of data
Owners
Cloud server

Fig. 3. Integrity verification schemes.

Data owners could perform integrity verification by themselves or delegate the task to trusted third parties. The basic framework of any integrity verification scheme consist of three participating parties: client, cloud storage server (CSS) and third party auditor (TPA). The client stores the data on cloud and the objective of TPA is to verify the integrity of data. The main life cycle of a remote integrity verification scheme consists of the following steps [29].

- Setup and data upload: In order to verify the data without retrieving the actual file, the client needs to prepare verification metadata. Metadata are computed from the original data and is stored alongside the original data. For practical use, the metadata should be smaller in size compared to the original dataset. The metadata are computed with the help of homomorphic linear authenticator (HLA) or Homomorphic verifiable tag (HVT). HLA or HVA have evolved from digital signatures like RSA and BLS (mathematical schemes to verify integrity of data). Each block stored on cloud is accompanied with an HVT or HLA tag. Current integrity verification methods also utilizes authenticated data structure like Merkel Hash Tree (MHT) [30]. MHT is similar to binary tree, each node will have maximum of two child nodes. MHT is a tree of hashes in which leaves are hashes of data blocks.
- Authorization for TPA: The TPA who can verify data from cloud server on data owner's behalf needs to be authorized by the data owner. There is also a security risk if the third party can ask for indefinite integrity proofs over certain dataset. This step is only required when client wants some third party to verify data.
- Challenge and verification of data storage: To verify the integrity of the data, a challenge message is sent to the server by TPA on client's behalf. The server will compute a response based on the challenge message and send it to TPA. The TPA can then verify the response to find whether the data are intact. The scheme has public verifiability if this verification can be done without the client's secret key. Most of the schemes, such as provable data processing (PDP) and proofs of retrievability (POR), support public data verification. We will discuss PDP and POR later. The major issue with public verification schemes is that it can enable malicious practices. For instance, the challenge message is very simple and everyone can send a challenge message to CSS for a proof of certain file block. A malicious user can launch a distributed denial of service (DDOS) attacks by sending multiple challenges from multiple clients by

causing additional overhead and congestion in network traffic.

- Data update: Data update occurs when some operations are performed on the data. The client needs to perform updates to some of the cloud data storage. Common could data update includes insert, delete, and modify operations.
- Metadata update: After some update operation is performed on the data, the client will need to update the metadata (HLA or HVT's) according with the existing keys. The metadata are updated in order to keep the data storage verifiable without retrieving all the data.
- Verification of updated data: Client also needs to verify if the data update is processed correctly or not as the cloud cannot be fully trusted. This is an essential step to ensure that the updated data still can be verified correctly in future.

*1) PDP:* Proposed by Ateniese *et al.* in 2007 [31]-[33], PDP scheme was built to offer block-less verification i.e., the verifier can verify the integrity of a proportion of the outsourced file through verifying a combination of pre-computed HVT or HVL. The HVT tags are used as the building block of PDP schemes and the tag construction is based on RSA signatures. The tag is stored on server together with the file and act as a verification metadata for the file block. The HVTs are unforgeable and have the following properties

- Block-less verification. Using HVTs, the server can construct a proof that allows the client to verify if the server possesses certain file blocks, even when the client does not have access to the actual file blocks.
- Homomorphic tags. Given the values of two homomorphic tags for two data blocks, adding the two homomorphic tags will correspond to the sum of the messages of both blocks.

*2) POR:* PORs are cryptographic proofs that enable a cloud provider to prove that a user can retrieve a targeted file in its entirety. POR consist basically of a challenge-response protocol in which the service provider proves to the user that the file is still intact and retrievable. The concept of POR and its first model was proposed by Jules *et al.* [32]. Unfortunately, this scheme can only be applied to static data storage such as an archive or library. Later in 2008, Shacham *et al.* [35] proposed an improved version of POR. They proposed a construction for private verification so that the data can only be verified with the secret key. As a result, no other party can verify it except for the client. The scheme was efficient because it admits short response and fast computation. Armknecht *et al.* [34] proposed outsourced proofs of retrievability (OPOR), in which users can task an external auditor to perform and verify POR with the cloud provider.

*3) Public Auditing:* Data integrity verification performed by third parties is termed as public auditing [36]-[37]. Wang *et al.* [36] proposed a scheme based on BLS signature that can support public auditing and full data dynamics, which is one of the latest works on public data auditing. However, this scheme lacks the support for fine-grained update (an operation which is applied to a smaller set from a large dataset such as a single row) and authorized auditing. Liu *et al.* [28] proposed a public auditing scheme with support of fine-grained updates over variable-sized file blocks. In addition, an authentication process between the client and TPA is also proposed to prevent TPA from endless challenges, thereby cutting the possibility of attacks over multiple challenges.

The problem with public auditing scheme is that the linear combination of blocks aggregated for assured auditing may reveal user information, especially if enough number of linear combination of the same blocks are collected. Wang *et al.* proposed a privacy preserving public auditing scheme [37]. When computing integrity proof, a random masking technique is incorporated to prevent the part of original file being extracted from several integrity proofs over this specific part of data.

## V. Privacy preserving in data processing

Privacy protection in data processing part can be divided into two phases. In the first phase, the goal is to safeguard information from unsolicited disclosure because the collected data may contain sensitive information about the data owner. In the second phase, the goal is to extract meaningful information from the data without violating the privacy. We will discuss the two phases in this section.

### A. PPDP

During PPDP, the collected data may contain sensitive information about the data owner. Directly releasing the information for further processing may violate the privacy of the data owner, hence data modification is needed in such a way that it does not disclose any personal information about the owner. On the other hand, the modified data should still be useful, not to violate the original purpose of data publishing. The privacy and utility of data are inversely related to each other and will be discussed in detail later in this section. Many studies have been conducted to modify the data before publishing or storing them [38]-[39] for further processing. To preserve the privacy of a user, PPDP mainly uses anonymization techniques. The original data are assumed to be sensitive and private and consist of multiple records. Each record may consist of the following four attributes [38].

- Identifier (ID): The attributes which can be used to uniquely identify a person e.g., name, driving license number, and mobile number etc.
- Quasi-identifier (QID): The attributes that cannot uniquely identify a record by themselves but if linked with some external dataset may be able to re-identify the records. An example of it is shown in Fig. 4.
- Sensitive attribute (SA: The attributes that a person may want to conceal e.g., salary and disease.
- Non-sensitive attribute (NSA): Non-sensitive attributes are attributes which if disclosed will not violate the privacy of the user. All attributes other than identifier, quasi-identifier and sensitive attributes are classified as non-sensitive attributes.

The data are anonymized by removing the identifiers and modifying the quasi-identifiers before publishing or storing for further processing. As a result of anonymization, identity

TABLE II
COMPARISON OF INTEGRITY VERIFICATION SCHEMES

| Integrity verification scheme | Features | Limitations |
|---|---|---|
| PDP | • Secure for remote data verification<br>• Based on Homomorphic verifiable tags<br>• Works well with static data | • Lack of privacy preserving support for TPA<br>• Insecure in dynamic environment due to replay attacks |
| POR | • POR guarantees correct data possession<br>• Error correcting codes (ECC) are used to recover corrupted blocks | • Only support limited number of challenging queries<br>• Auditing is difficult for dynamic data due to ECC |
| Public auditing | • Auditing is done by a third party<br>• Use BLS signatures to generate authentication values<br>• The scheme is proved to be secure | • Some information is leaked to auditor in the verification process |

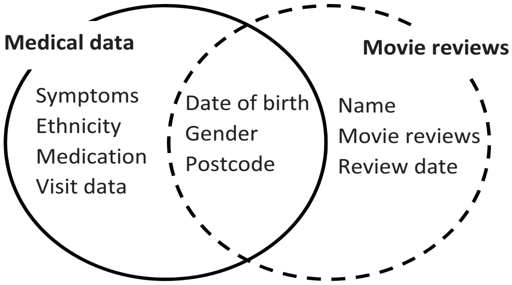

Fig. 4. QIDs and linking records.

of the data owner and sensitive values are hidden from the adversaries. How much data should be anonymised mainly depends on how much privacy we want to preserve in that data. The privacy models are basically classified into two categories based on the ability of an attacker to identify an individual [38]. The first category is based on the assumption that the attacker is able to identify the records of a specific user by linking the records with external data sources. The second category is based on the assumption that the attacker has enough background knowledge to conduct probabilistic attacks i.e., the attacker is able to make a confident guess about whether the specific user's record exists in the database or not.

There are several models proposed to deal with the above problems. Some of them include k-anonymity to prevent the record linkage, l-diversity to prevent attribute linkage and record linkage, t-closeness to prevent probabilistic attacks and attribute linkage [38].

*1) Anonymization Techniques:* Before publishing, the original table is modified according to the specified privacy requirements. To preserve the privacy, one of the following anonymization operations are applied to the data [38].

- Generalization : Generalization works by replacing the value of specific QID attributes with less specific description. In this operation some values are replaced by a parent value in the taxonomy of an attribute. An example of it can be representing a job attribute with artist instead of singer or actor. The types of generalization techniques include full domain generalization, subtree generalization, multidimensional generalization, sibling generalization, and cell generalization.
- Suppression: In suppression, some values are replaced with a special character (e.g., "*"), which indicates that a replaced value is not disclosed. Example of suppression schemes include record suppression, value suppression, and cell suppression.
- Anatomization: Instead of modifying the quasi-identifier or sensitive attributes, anatomization works by de-associating the relationship between the two. In this method, the data on QID and SA are released in two separate tables. One table contains quasi-identifier and the other table contains sensitive attributes. Both tables contain one common attribute which is often called GroupID. The same group will have the same value for GroupID linked to the sensitive values in the group.
- Permutation: In permutation, the relationship between quasi-identifier and numerically sensitive attribute is de-associated by partitioning a set of records into groups and shuffling their sensitive values within each group.
- Perturbation: In perturbation, the original data values are replaced by some synthetic data values, so that the statistical information computed from modified data does not differ significantly from the statistical information computed from the original data. Some examples include adding noise, swapping data, and generating synthetic data. The problem with perturbation is that the published record are synthetic and does not mean anything in the real world and hence is meaningless to the recipients. They only preserve the statistical properties explicitly selected by the publisher.

*2) Privacy-utility trade-off:* A high level of data anonymization indicates that the privacy is well protected. However, on the other hand, it may also affect the utility of the data, which means that less values can be extracted from the data. Therefore, balancing the trade-off between privacy and utility is very important in big data applications. The reduction in data utility is represented by information loss. Various methods have been proposed in the literature for measuring the information loss, some of the examples include minimal distortion [40], discernibility metric [41], the normalized average equivalence class size metric [42], weighted certainty penalty [43], and information theoretic metrics [44]-[45]. To solve the problems of trade-off between privacy and utility, PPDP algorithms usually take greedy approach to achieve proper trade-off. These algorithms work by generating multiple tables using the given metrics of privacy preservation and information loss, all of which satisfy the requirement of specific privacy model during the anonymization process. Output of the greedy algorithm is the table with minimum information loss.

Quantifying privacy is a very hard task. For example, consider a scenario where a piece of data is collected from a data owner. The data owner is free to decide how much and what kind of information he or she wants to share with a third party. Once the data are handed over to the third party, some privacy loss may occur. Different data owners may provide the same data to the third party. However, when privacy disclosure happens, some individuals who treat privacy seriously may perceive more loss than those who have little concern about the privacy.

### *B. Extracting knowledge from data*

To extract useful information from big data without breaching the privacy, privacy preserving data mining techniques have been developed to identify patterns and trends from data. Those techniques cannot be applied straightaway to big data as big data may contain large, complex and dynamically varying data. To handle big data in an efficient manner, those techniques should be modified, or some special set of techniques should be used. In addition to this, those modified techniques should address the privacy concern. There are several techniques proposed to analyze large-scale and complex data. These techniques can be broadly grouped into clustering, classification and association rule based techniques.

*1) Privacy preserving clustering:* Clustering is one of the popular data processing techniques for its capability of analyzing un-familiar data. The fundamental idea behind clustering is to separate unlabelled input data into several different groups [46]. Conventional clustering algorithms require data to be in the same format and be loaded into a single processing unit, which is not suitable for big data processing. Many solutions [47]-[48] have been presented in the recent decade. However, due to the nature of the big data, they have several disadvantages, among which computational complexity and privacy concern are the major problems. To handle the issue of computational complexity, in [49], Shirkhorshidi *et al.* introduced sampling and dimension reduction solutions for single-machine clustering and parallel and map-reduce solutions for multiple-machine clustering. To improve the efficiency, in [51], cloud computing based parallel processing was proposed. To make clustering feasible for very large data sets, in [53], Feldman *et al.* presented a parallel processing approach in which core sets are created using a tree construction. Compared to traditional clustering algorithms, in [53], the processing time and the required amount of energy are significantly reduced. Nevertheless, in all of these methods [47]-[53], privacy is a major concern. Privacy preservation in clustering is a challenging problem when large volume complex data are involved. In the early days, hybrid geometric data transformation based methods [54]-[55] were proposed to protect the privacy in clustering. However, these methods alter numerical attributes by translations, scaling and rotations. Although certain level of privacy could be achieved, data utility is usually reduced. Thus these methods are not practically feasible. In [56], Oliveira and Zaiane proposed a method for centralized data by using dimensionality reduction and object similarity based representation. Since this method is specifically designed for centralized data, it cannot be used with more commonly existing de-centralized big data. To improve the efficiency of clustering in new data (non-familiar), in [50], privacy-preserving clustering based on the probability distributed model was proposed. In order to handle complex and distributed data, in [52], a novel algorithm called distributed local clustering is presented. In [52], secure multi-party computation based techniques such as homomorphic encryption, are used to achieve privacy protection. In the above mentioned methods, clustering is done using low order statistics. When the input data are complex, these lower order statistics are inadequate and could yield poor clustering results. To overcome this, in [57], Shen and Li developed a clustering method using information theoretic measures as a cost function to develop a linear and a kernel distributed clustering algorithm. In [57], the nodes only exchange a few parameters instead of original data with their neighbors.

*2) Privacy preserving data classification:* Classification is a technique of identifying, to which predefined group a new input data belongs. Similar to clustering algorithm, classification algorithms are traditionally designed to work in centralized environments. To cope up with the demands of big data, traditional classification algorithms were modified to suit parallel computing environment. For example, in [58], a classification algorithm is designed to process data in two ways. This algorithm, known as "classify or send for classification", either classifies the data by themselves or forward the input data to another classifier. It is computationally efficient particularly when handling large and complex data. In another novel classification algorithm, Rebentrost *et al.* [59] proposed a quantum based support vector machine for big data classification. This method reduces the computational complexity and the required training data. The main limitation of this method is the immature hardware technologies in quantum computing. Even though the classification algorithms developed for big data can reach a reasonable level of performance, these algorithms do not pay much attention to the data privacy either. In [60], Agrawal *et al.* proposed a privacy preserving classification algorithm for the discovery for knowledge from the data. The original data are altered by adding random offsets. Then Bayesian formula is used to derive the density function of the original data in order to reconstruct the decision tree. The major problem with this method is it is only suitable for the centralized data. In [61], another privacy preserving data mining algorithm is proposed using random reconstruction techniques. The random operation in the algorithm protects the privacy of the original data via data scrambling. However, this method is also not suitable for diverse data. Unlike the methods in [60] and [61], a privacy preserving method is proposed in [62] for distributed databases. In this work, a random perturbation matrix is used to protect the privacy of the data. Due to the nature of the algorithm, it requires the reconstruction of the original data set from the altered data set. This significantly reduces the accuracy of the algorithm. To improve the accuracy, the authors in [63] developed an algorithm using single-attribute data random matrix. This matrix is used to slightly modify the data, and the reconstruction of original data set is improved by the use of multi-attribute joint distribution matrix. This method improves the accuracy at the expense of privacy. By using the advantage of multi-attribute joint distribution matrix, in [64], Zhang and Bi proposed a privacy preserving method for classification with slightly improved the accuracy and privacy, but this method is incapable of handling large and complex data.

*3) Privacy preserving association rule mining:* While clustering and classification try to group the input data, association rules are designed to find the important relationships or patterns between the input data. Finding the relationships on larger data set has been studied for many years. In the early days, tree structures such as FP-tree [65] were used in finding the pattern. Early algorithms were not suitable for large and diverse data, because parallel computing and cloud computing technologies are used in these scenarios. To handle large and complex data in an efficient way, several methods [66]-[68] have been developed using map-reduce. The map-reduction concept is ideally suitable for could based association rule finding algorithms. However, the association rule mining methods proposed in [66]-[68] do not consider the privacy of the input data. Protecting privacy in association rule mining is an operation to protect the sensitive information from being mined. For example in [69], privacy is preserved by distorting the original data. In [69], data are distorted in such a way that the distorted data can be used to generate an approximation of the original data distribution, without exposing the values in the original data. In this approach, the level of privacy is relatively low. Thus, to enhance the privacy, in [70], tougher conditions are imposed to reduce the privacy leakage. Recently in [71] and [72], privacy protection techniques were applied to Boolean association rules. Similar to other methods, the original data are also distorted in these works. In some methods, cryptographic techniques are used to construct the decision trees [73]. In [73], privacy-preserving data mining is considered as a part of secure multi-party computation. Although these methods achieve some level of privacy and accuracy, they are not fully capable of handling large and complex data.

## VI. Conclusion and Future Research challenges

The amount of data are growing everyday and it is impossible to imagine the next generation applications without producing and executing data driven algorithms. In this paper, we have conducted a comprehensive survey on the privacy issues when dealing with big data. We have investigated privacy challenges in each phase of big data life cycle and discussed some advantages and disadvantages of existing privacy preserving technologies in the context of big data applications. A lot of works have been done to preserve the privacy of users from data generation to data processing, but there still exist several open issues and challenges. In this section, we discuss a few future research directions for big data privacy.

### A. Access control and secure end to end communication

To ensure that the data are only accessible by authorized users and for end to end secure transfer of data, access control methods and different encryption techniques like IBE, ABE, and PRE, are used. The main problem of encrypting large datasets using existing techniques is that we have to retrieve or decrypt the whole dataset before further operations could be performed. These techniques does not allow data owners to easily perform fine grained actions such as sharing records for data analytics. Techniques such as PRE have solved this problem up to some extend. However, to obtain the values from the data, sometimes the data need to be shared multiple times with different companies. As different companies have different cryptographic keys, the data need to be decrypted and then re-encrypted again which not only has a computational overhead but also has a possibility of data leakage. To solve these kind of problems, we need encryption techniques which allows data sharing between different parties without decrypted and re-encrypting process.

### B. Data anonymization

Data is anonymized by removing the personal details to preserve the privacy of users. It indicates that it would not be possible to identify an individual only from the anonymized data. However, due to the availability of huge volumes of data and powerful data analytic tools, the existing anonymization techniques are becoming increasingly ineffective. In big data scenarios, anonymization needs to be more than just masking or generalizing certain fields. One needs to carefully analyse if the anonymized data are vulnerable to any attacks. For that, we need to study different attack models and information loss metric for big data anonymization. Moreover, most of the existing anonymization techniques are for static data, while much practical data is dynamic. Thus, we need to propose new privacy and utility metrics. Furthermore, data anonymization is a cumbersome process and it needs to be automated to cope with the growing 3 V's.

### C. Decentralized storage

As our personal data are gradually collected and stored on centralized cloud server over the time, we need to understand the associated risk regarding privacy. The concept of centralized collection and storage of personal data should be challenged. In centralized storage, a single point of failure would indicate the lost of the whole data. One flaw or one breach in privacy can lead to a devastating consequences, which is happening more frequently with sophisticated methods of attacks. Instead of centralizing all the computation, we can bring the computation to intelligent agents running on our own personal devices. Using such schemes, business models can still be profitable and we can regain our privacy by hosting our data in personal encrypted clouds. There are researchers who are strongly suggesting to adopt decentralized storage [74]. Some works have been done with projects like *OwnCloud* and the *IndieWeb* [74]. To adopt the view of data distribution, we need algorithms that are capable to work over extreme data distribution and build models that learn in a big data context.

### *D. Effective machine learning techniques and distributed data analytics*

Machine learning and data mining should be adapted to unleash the full potential of collected data. Nowadays, machine learning techniques, together with the improvement of computational power (e.g., cloud computing), have come to play a vital role in big data analytics. They are employed widely to leverage the predictive power of big data. For example, the predictive power of big data is extensively used in medical science and astronomy. Most of these computations are done by third party resources on private data, which can pose a threat to the privacy of users. To protect privacy, machine learning algorithms such as classification, clustering and association rule mining need to be deployed in a privacy preserving way.

Sometimes the data owned by an organization (e.g., hospitals) does not have sufficient information to discover useful knowledge in that domain, and acquiring that data may be costly or difficult due to legal constraints and fear of privacy violation. To solve such problems, we need to design privacy preserving distributed analytic systems which are able to process different datasets from different organizations while preserving the privacy of each dataset.

Secure multiparty computation techniques such as homomorphic encryption can be deployed to solve such issues. The main challenge in deploying homomorphic encryption in the context of big data analytics is to keep the computational complexity as low as possible.

## References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, “Big data: The next frontier for innovation, competition, and productivity,” *Mickensy Global Institute* , pp. 1–137, Jun. 2011.

[2] B. Matturdi, X. Zhou, S, Li, and F. Lin, “Big data security and privacy: A review,” *China Communications*, vol. 11, no. 14, pp. 135–145, Apr. 2014.

[3] J. Gantz and D. Reinsel, “Extracting value from chaos,” in *Proc. on IDC IView*, Jun. 2011, pp. 1–12.

[4] A. Katal, M. Wazid, R. H. Goudar, “Big data: Issues, challenges, tools and good practices,” in *Proc. of IEEE Int. Conf. on Contemporary Computing* , Aug. 2013, pp. 404–409.

[5] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, “Information security in big data: Privacy and data mining,” in *IEEE Access*, vol. 2, pp. 1149–1176, Oct. 2014.

[6] H. Hu, Y. Wen, T. S. Chua, and X. Li, “Toward scalable systems for big data analytics: A technology tutorial,” *IEEE Access* , vol. 2, pp. 652–687, Jul. 2014.

[7] Z. Xiao and Y. Xiao, “Security and privacy in cloud computing,” *IEEE Trans. on Communications Surveys and Tutorials* , vol. 15, no.2, pp. 843–859, May 2013.

[8] H. Cheng, C. Rong, K. Hwang, W. Wang, and Y. Li, “Secure big data storage and sharing scheme for cloud tenants,” *China Communications*, vol. 12, no. 6, pp. 106–115, Jun. 2015.

[9] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” *IEEE Trans. on Parallel and Distributed Systems* ,vol. 25, no. 1, pp. 222–233, Jan. 2014.

[10] O. M. Soundararajan, Y. Jenifer, S. Dhivya, and T. K. P. Rajagopal, “Data security and privacy in cloud using RC6 and SHA algorithms,” *Networking and Communication Engineering*, vol. 6, no. 5, pp. 202–205, Jun. 2014.

[11] S. Singla and J. Singh, “Cloud data security using authentication and encryption technique,” *Global Journal of Computer Science and Technology*, vol. 13, no. 3, pp. 2232–2235, Jul. 2013.

[12] U. Troppens, R. Erkens, W. Mueller-Friedt, R. Wolafka, and N. Haustein, *Storage networks explained: Basics and application of fibre channel SAN, NAS, ISCSI, infiniband and FCoE. New York, NY, USA*, Wiley, 2011.

[13] P. Mell and T. Grance, “The NIST definition of cloud computing,” *National Institute of standards and technology*, vol. 53, no. 6, pp. 50, 2009.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted Data,” in *Proc. of ACM Conf. on Computer and Communication Security* , Oct. 2006, pp. 89–98.

[15] J. Bethencourt, A. Sahai, and B. Waters,“Ciphertext-policy attribute based encryption,”in em Proc. of IEEE Int. Conf. on Security and Privacy, May 2007, pp. 321–334.

[16] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, “DAC-MACS: Effective data access control for multiauthority cloud storage systems,” *IEEE Trans. on Information Forensics Security*, vol. 8, no. 11, pp. 1790–1801, Oct. 2013.

[17] K. Yang and X. Jia, “Expressive, efficient, and revocable data access control for multi-authority cloud storage,” *IEEE Trans. on Parallel Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, Jul. 2014.

[18] K. Yang, X. Jia, and K. Ren, “Secure and verifiable policy update outsourcing for big data access control in the cloud,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, Nov. 2015.

[19] X. Boyen and B. Waters, “Anonymous hierarchical identity-based encryption,” *Advances in Cryptology - ASIACRYPT*, vol. 4117, pp. 290–307, Aug. 2006.

[20] M. Mambo and E. Okamoto, “Proxy cryptosystems: Delegation of the power to decrypt ciphertexts,” *Fundamentals of electronics, communications and computer sciences*, vol. 80, no. 1, pp. 54–63, 1997.

[21] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” *Advances in Cryptology - ASIACRYPT* , pp. 127-144, 1998.

[22] M. Green and G. Ateniese, “Identity-based proxy re-encryption,” in *Proc. of Int. Conf. on Applied Cryptography and Network Security* , vol. 63, no. 1, Jan. 2007, pp. 288–306.

[23] J. Shao, “Anonymous ID-based proxy re-encryption,” in *Proc. of Int. Conf. on Information Security and Privacy* , vol. 7372, Jul. 2012, pp. 364–375.

[24] K. Liang, W. Susilo, and J. Liu, “Privacy-preserving ciphertext sharing mechanism for big data Storage,” *IEEE Trans. on Information Forensics and Security* , vol. 10, no. 8 , pp. 1578–1589, Aug. 2015.

[25] C. Gentry, “A fully homomorphic encryption scheme,” *Ph.D dissertation, Stanford University*, California, USA, 2009.

[26] S. Nepal, R. Ranjan, and K. K. R. Choo, “Trustworthy processing of healthcare big data in hybrid clouds,” *IEEE Trans. on Cloud Computing*, vol. 2, no. 2, pp. 78–84, Jun. 2015.

[27] X. Huang, and X. Du, “Achieving big data privacy via hybrid cloud,” in *Proc. of Int. Conf. on INFOCOM* , April. 2014, pp. 512–517.

[28] C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and K. Rao, “Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2234–2244, Aug. 2014.

[29] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos, and J. Chen, “Public auditing for big data storage in cloud computing - a survey,” in *Proc. of IEEE Int. Conf. on Computational Science and Engineering* , Dec. 2013, pp. 1128–1135.

[30] R. C. Merkle, “A digital signature based on a conventional encryption function,” *Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, pp. 369–378, Jan. 1988.

[31] G. Ateniese, R. B. Johns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *Proc. of Int. Conf. of ACM on Computer and Communications Security*, 2007, pp. 598–609.

[32] A. Juels and B. S. Kaliski, “PORs: proofs of retrievability for large files,” in *Proc. of ACM Conf. on Computer and Communication Security*, Oct. 2007, pp. 584–597.

[33] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," *Information and System Security*, vol. 14, no. 1, p. Article 12, May 2011.

[34] F. Armknecht, J. M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in *Proc. of ACM Conf. on Computer and Communications Security*, Nov. 2014, pp. 831–843.

[35] H. Shacham, B. Waters, "Compact Proofs of Retrievability," *Advances in Cryptology - ASIACRYPT*, pp. 90–107, Dec. 2008.

[36] Q. Wang, C. Wang, K. Ren, W. Lou, and J .Li, "Enabling public auditability and data dynamics for storage security in cloud computing, *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, May 2011.

[37] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. of IEEE Int. Conf. on INFOCOM*, 2010, pp. 1–9.

[38] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, Jun. 2010, pp. 14.

[39] R. C. W. Wong and A. W. C. Fu, "Privacy-preserving data publishing: An overview," *Synthesis Lectures Data Management* , vol. 2, no. 1, pp. 1–138, 2010.

[40] L. Sweeney, "k -anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[41] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. of Int. Conf. on data engineering*, Apr. 2005, pp. 217–228.

[42] K. Lefevre, D. J. Dewitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proc. of Int. Conf. on data engineering*, Apr. 2006, pp. 25.

[43] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W. C. Fu, "Utility-based anonymization for privacy preservation with less information loss," *ACM SIGKDD Explorations Newsletter* , vol. 8, no. 2, pp. 21–30, Dec. 2006.

[44] A. Gionis and T. Tassa, "k-anonymization with minimal loss of information,"*IEEE Trans. on Knowledge and Data Engneering*, vol. 21, no. 2, pp. 206–219, Feb. 2009.

[45] L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. J Liu, "Privacy or utility? a contract theoretic approach to data collection," *IEEE Trans. on Signal processing*, vol. 9. No. 7, pp.1256–1269, Sep. 2015.

[46] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 21, no. 3, pp. 264–323, Sep. 1999.

[47] R. Xu and D. Wunsch, "Clustering," *Wiley-IEEE*, 2009.

[48] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: taxonomy and empirical analysis," *IEEE Trans. Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, Jun. 2014.

[49] A. S. Shirkhorshidi, S. R. Aghabozorgi, Y. W. Teh, and T. Herawan, "Big data clustering: a review," in *Proc. Int. Conf. on Computational Science and Its Applications*, 2014, pp. 707–720.

[50] W. Xiao-dan, Y. Dian-min, L. Feng-li, and C. Chao-hsien, "Distributed model based sampling technique for privacy preserving clustering," in *Proc. Int. Conf. on Management Science and Engineering*, 2007, pp. 192–197.

[51] H. Xu, Z. Li, S. Guo, and K. Chen, "Cloudvista: interactive and economical visual cluster analysis for big data in the cloud," in *Proc. VLDB Endowment*, 2012, pp. 1886–1889.

[52] A. M. Elmisery and H. Fu, "Privacy preserving distributed learning clustering of healthcare data using cryptography protocols," in *Proc. IEEE 34th Annual. Computing Software and Application Conf. Workshops*, 2010, pp. 140–145.

[53] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: constant-size coresets for k-means, pca and projective clustering," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2013, pp. 1434–1453.

[54] S. R. M. Oliveira and 0. R. Zaiane, "Privacy preserving clustering by data transformation," in *Proc. 18th Brazilian Symposium on Databases*, 2003, pp. 304–318.

[55] S. R. M. Oliveira and 0. R. Zaiane, "Geometric data transformation for privacy preserving clustering," *Technical Report TR 03-12*, 2003.

[56] S. R. M. Oliveira and O. R. Zaiane, "Privacy preserving clustering by object similarity-based representation and dimensionality reduction transformation," in *Proc. ICDM Workshop on Privacy and Security Aspects of Data Mining*, 2004, pp. 40–46.

[57] P. Shen and C. Li, "Distributed information theoretic clustering," *IEEE Trans. on Signal Process*, vol. 62, no. 13, pp. 3442–3453, Jul. 2014.

[58] C. Tekin and M. V. D. Schaar, "Distributed online big data classification using context information," in *Proc. Int. Conf. on Communication, Control, and Computing*, 2013, pp. 1435–1442.

[59] P. Rebentrost, M. Mohseni, and S. Lloyd. 2014. *Quantum support vector machine for big feature and big data classification* [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1307.html#RebentrostML13.

[60] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 439–450, 2000.

[61] A. Evmievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proc. ACM symposium on Principles of database systems*, 2003, pp. 211–222.

[62] S. Agrawal and J. R. Haritsa, "A framework for high accuracy privacy-preserving mining," in *Proc. 21st Int. Conf. on Data Engineering*, 2005, pp. 193–204.

[63] G. Weiping, W. Wei, and Z. Haofeng, "Privacy preserving classification mining," *Journal of Computer Research and Development*, vol. 43, no. 1, pp. 39-45, 2006, pp. 39–45.

[64] X. Zhang and H. Bi, "Research on privacy preserving classification data mining on random peturbation," in *Proc. Int. Conf. on Information Networking and Automation*, pp. 173–178, 2010.

[65] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2000. pp. 1–12.

[66] M. Y. Lin, P. Y. Lee, and S. C. Hsueh, "A priori-based frequent item set mining algorithms on map reduce," in *Proc. Int. Conf. on Ubiquitous Information Management and Communication*, 2012, pp. 76.

[67] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, "PARMA: a parallel randomized algorithm for approximate association rules mining in mapreduce," in *Proc. ACM Int. Conf. on Information and Knowledge Management*, 2012, pp 85–94.

[68] C. S. Leung, R. MacKinnon, and F. Jiang, Reducing the search space for big data mining for interesting patterns from uncertain data, in *Proc. Int. Conf. on Big Data*, 2014, pp 315–322.

[69] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. 2000 ACM SIGMOD Conf. Management of Data*, pp. 439–450, 2000.

[70] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proc. 20th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems*, 2001, pp. 247–255.

[71] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proc. Eighth ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2002, pp. 217–228.

[72] S.J. Rizvi and J.R. Haritsa, "Maintaining data privacy in association rule mining," in *Proc. of 28th Int. Conf. on Very Large Databases*, 2002, pp. 682–693.

[73] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Advances in Cryptology*, 2000, pp. 36–54.

[74] Z. H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," *IEEE Trans. on Computational Intelligence Magazine*, vol. 9, no. 4, pp.62–74, Nov. 2014.